UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,262	11/25/2003	Bjorn-Harald Sjogren	115719	4105

29078 7590 03/17/2008

CHRISTIAN D. ABEL
ONSAGERS AS
POSTBOKS 6963 ST. OLAVS PLASS
NORWAY, N-0130
NORWAY

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	**PAPER NUMBER**
2168	
MAIL DATE	**DELIVERY MODE**
03/17/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/720,262	SJOGREN ET AL.	
	Examiner	Art Unit	
	MAHESH H. DWIVEDI	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *31-33,35-38,40-43 and 45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *31-33,35-38,40-43 and 45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. Receipt of Applicant's Amendment filed on 12/20/2007 is acknowledged. The amendment includes the cancellation of claims 19-30, 34, 39, and 44, and the amending of claims 31, 36, and 41.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 31-33, 36-38, and 41-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Jacobs et al.** (U.S. Patent 6,105,025) in view of **Haderle et al.** (U.S. Patent 4,933,848).

5. Regarding claims 31 and 41, **Jacobs** teaches a constraint enforcer and database system comprising:

A) a set of constraints that governs the integrity of information stored in the data system (Column 7, lines 1-9, 43-64);

B) said enforcer being arranged to delay constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on

the check stack at the end of the transaction (Column 7, lines 1-9, 43-64, Column 9, lines 17-29, Column 10, lines 11-19);

C) the constraint enforcer comprising: a stack maker module, arranged for creating and updating said check stack (Column 9, lines 17-29, Column 10, lines 13-31);

D) said stack maker module being operatively connected to a runtime module in the database system (Column 6, lines 48-51, Column 9, lines 17-29); and

E) arranged to receive data from said runtime module (Column 6, lines 48-51, Column 9, lines 17-29);

F) an enforcer module (Column 7, lines 1-9, 43-64);

G) arranged to receive check data from the check stack (Column 7, lines 1-9, 43-64);

H) to process the check data received from the check stack (Column 7, lines 1-9, 43-64); and

I) to provide resulting data to the runtime module (Column 6, lines 48-51, Column 9, lines 17-29);

J) a conceptual rules module wherein said constraints are stored in the form of rules for prescribing permitted states and transitions that the database can undertake (Column 2, lines 65-67-Column 3, lines 1-16, Column 7, lines 1-6, Column 10, lines 15-17);

K) the conceptual rules module being operatively connected to said stack maker (Column 6, lines 48-51, Column 9, lines 17-29);

L) said stack maker module is arranged to retrieve constraints from said conceptual rules module (Column 7, lines 1-6, Column 9, lines 17-29, and Column 10, lines 13-31).

The examiner notes that **Jacobs** teaches **"a set of constraints that governs the integrity of information stored in the data system"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6), "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"said enforcer being arranged**

to delay constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction" as enforcement may be deferred until processing is completed for either a statement or transaction" (Column 7, lines 46-47), "constraint enforcement is deferred until the end of transaction (i.e., transaction level enforcement" (Column 7, lines 57-58). The examiner further notes that **Jacobs** teaches **"the constraint enforcer comprising: a stack maker module, arranged for creating and updating said check stack"** as "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19), "insert, delete, and update operations" (Column 10, line 13), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"said stack maker module being operatively connected to a runtime module in the database system"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"arranged to receive data from said runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"an enforcer module"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-3). The examiner further notes that **Jacobs** teaches **"arranged to receive check data from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"to process the check data received from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-

3) and "if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement)...are examined as an initial part of transaction commit processing" (Column 7, lines 57-62). The examiner further notes that **Jacobs** teaches **"to provide resulting data to the runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"a conceptual rules module wherein said constraints are stored in the form of rules for prescribing permitted states and transitions that the database can undertake"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17), and "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"the conceptual rules module being operatively connected to said stack maker"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"said stack maker module is arranged to retrieve constraints from said conceptual rules module"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs.

Jacobs does not explicitly teach:

M) wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the begging and end of the transaction.

Haderle, however, teaches **"wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the begging and end of the transaction"** as "The major phases of the preferred embodiment of this method are shown schematically in FIG. 3. In the Data Load phase 24, all input data rows or records are placed in their target tables 10, 12 without regard to referential integrity constraints. Foreign key values and control information are extracted during the Data Load phase 24 to allow constraint checking to be performed after the tables 10, 12 have been loaded. Index key values are also extracted during the Data Load phase 24. In the Sort phase 26, the key values and attached control information are sorted to allow an optimal order of index updating and constraint enforcement. In the Index Update phase 28, the primary indexes 22 are updated to include the new input data, using the sorted key values from the Sort phase 26. In the Enforcement phase 30, the sorted foreign key values are evaluated against the contents of the appropriate primary indexes 22 to determine if any input data rows introduced by the Data Load phase 24 cause referential integrity constraint violations. In the Discard phase 32 rows causing referential constraint violations, and new rows which are descendents of rows causing such violations are deleted. At the end of the Discard phase 32 the tables 10, 12 again possess referential integrity. Finally, the Report phase 34 prepares a summary report 36 of the rows deleted during the Discard phase 32" (Column 3, lines 8-34).

The examiner notes that **Haderle's** method of inputting the data into target tables [("all input data rows or records are placed in their target tables 10, 12 without regard to referential integrity constraints") (i.e. allowing a transaction to occur)], checking for constraint violations [("In the Enforcement phase 30, the sorted foreign key values are evaluated against the contents of the appropriate primary indexes 22 to determine if any

input data rows introduced by the Data Load phase 24 cause referential integrity constraint violations")(i.e. allowing the conceptual rules to be broken during the transaction)], and correcting those violations so that the database is in a consistent state(i.e. no constraints) [("rows causing referential constraint violations, and new rows which are descendents of rows causing such violations are deleted")] teaches the aforementioned limitation.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Haderle's** would have allowed **Jacobs's** to provide a method for efficient processing of database constraints so that the order of transactions in terms of constraints are processed correctly, as noted by **Haderle** (Column 1, lines 44-51).

Regarding claims 32, 37, and 42 **Jacobs** further teaches a constraint enforcer, method, and database system comprising:
A) wherein said check stack is stored on persistent or volatile memory. (Column 10, lines 13-31, Column 16, lines 51-53, Figure 6).

The examiner notes that **Jacobs** teaches **"wherein said check stack is stored on persistent or volatile memory"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17) and "The present invention can be implemented on a general purpose computer such as illustrated in FIG. 6…FIG. 6 also includes a video memory 614, main memory 615 and mass storage 612, all coupled to bi-directional system bus 618" (Column 16, lines 46-53).

Regarding claims 33, 38, and 43 **Jacobs** further teaches a constraint enforcer, method, and database system comprising:
A) wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by an insert operator (Column 10, lines 11-17, Figures 3A-3B).

The examiner notes that **Jacobs** teaches **"wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by**

an insert operator" as "Examples of insert, delete, and update operations using a uniqueness-required index are provided with reference to FIGS. 3A-3B" (Column 10, lines 13-15).

Regarding claim 36, **Jacobs** teaches a method comprising:

A) a set of constraints that governs the integrity of information stored in the data system (Column 7, lines 1-9, 43-64);

B) the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake (Column 2, lines 65-67-Column 3, lines 1-16, Column 7, lines 1-6, Column 10, lines 15-17);

C) said method comprising the steps of: delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction (Column 7, lines 1-9, 43-64, Column 9, lines 17-29, Column 10, lines 11-19);

D) by a stack maker module operatively connected to a runtime module in said database system (Column 6, lines 48-51, Column 9, lines 17-29);

E) receiving data from said runtime module (Column 6, lines 48-51, Column 9, lines 17-29);

F) creating and updating said check stack (Column 9, lines 17-29); and

G) retrieving constraints from said conceptual rules module (Column 7, lines 1-6, Column 9, lines 17-29, and Column 10, lines 13-31); and

H) by an enforcer module: receiving check data from the check stack (Column 7, lines 1-9, 43-64);

I) processing the check data received from the check stack (Column 7, lines 1-9, 43-64); and; and

J) providing resulting data to the runtime module (Column 6, lines 48-51, Column 9, lines 17-29).

The examiner notes that **Jacobs** teaches **"a set of constraints that governs the integrity of information stored in the data system"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new

data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6), "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19), and "Uniqueness required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17). The examiner further notes that **Jacobs** teaches **"the constraints being stored in a conceptual rules module in the form of rules for prescribing permitted states and transitions that the database can undertake"** as "Uniqueness-required index 308 is a B-tree structured index created on column 302N" (Column 10, lines 15-17), and "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"said method comprising the steps of: delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction"** as enforcement may be deferred until processing is completed for either a statement or transaction" (Column 7, lines 46-47), "constraint enforcement is deferred until the end of transaction (i.e., transaction level enforcement" (Column 7, lines 57-58). The examiner further notes that **Jacobs** teaches **"by a stack maker module operatively connected to a runtime module in said database system"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that **Jacobs** teaches **"receiving data from said runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"creating and updating said check**

stack" as "According to one embodiment of the invention, a list is generated for each uniqueness-required index for each session. Every time the counter of an index changes, an entry is added to the list associated with the index. The entry includes the new value of the non-uniqueness count of the index and a savepoint indicator. When a transaction is rolled back to a particular savepoint, entries are removed from the tail of the list until an entry is encountered that has a savepoint indicator that identifies the savepoint to which the transaction is being rolled-back. The entry that contains the corresponding savepoint-indicator will indicate the uniqueness count value that the corresponding index should have after the roll-back is performed" (Column 9, lines 17-29). The examiner further notes that **Jacobs** teaches **"retrieving constraints from said conceptual rules module"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs. The examiner further notes that **Jacobs** teaches **"by an enforcer module: receiving check data from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation. A constraint violation occurs when the new data value is inserted in a column having a uniqueness constraint and the value already exists in column" (Column 7, lines 1-6). The examiner further notes that **Jacobs** teaches **"processing the check data received from the check stack"** as "When a data value is created or added during a transaction or statement, a check is made to see if the new data vale has created a constraint violation" (Column 7, lines 1-3) and "if constraint enforcement is deferred until the end of a transaction (i.e., transaction level enforcement)...are examined as an initial part of transaction commit processing" (Column 7, lines 57-62). The examiner further notes that **Jacobs** teaches **"providing resulting data to the runtime module"** as "session (i.e., a connection between the application and the database system" (Column 6, lines 50-51) and "a list is generated

for each uniqueness-required index for each session" (Column 9, lines 17-19). The examiner further notes that it common knowledge that during a session, information and data is exchanged between application and database programs.

Jacobs does not explicitly teach:

K) wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the begging and end of the transaction.

Haderle, however, teaches **"wherein said constraints are constraints executed within the transaction which allow conceptual rules to be broken during the transaction, but allow the database system to be in a consistent state at the begging and end of the transaction"** as "The major phases of the preferred embodiment of this method are shown schematically in FIG. 3. In the Data Load phase 24, all input data rows or records are placed in their target tables 10, 12 without regard to referential integrity constraints. Foreign key values and control information are extracted during the Data Load phase 24 to allow constraint checking to be performed after the tables 10, 12 have been loaded. Index key values are also extracted during the Data Load phase 24. In the Sort phase 26, the key values and attached control information are sorted to allow an optimal order of index updating and constraint enforcement. In the Index Update phase 28, the primary indexes 22 are updated to include the new input data, using the sorted key values from the Sort phase 26. In the Enforcement phase 30, the sorted foreign key values are evaluated against the contents of the appropriate primary indexes 22 to determine if any input data rows introduced by the Data Load phase 24 cause referential integrity constraint violations. In the Discard phase 32 rows causing referential constraint violations, and new rows which are descendents of rows causing such violations are deleted. At the end of the Discard phase 32 the tables 10, 12 again possess referential integrity. Finally, the Report phase 34 prepares a summary report 36 of the rows deleted during the Discard phase 32" (Column 3, lines 8-34).

The examiner notes that **Haderle's** method of inputting the data into target tables [("all input data rows or records are placed in their target tables 10, 12 without regard to

referential integrity constraints") (i.e. allowing a transaction to occur)], checking for constraint violations [("In the Enforcement phase 30, the sorted foreign key values are evaluated against the contents of the appropriate primary indexes 22 to determine if any input data rows introduced by the Data Load phase 24 cause referential integrity constraint violations")(i.e. allowing the conceptual rules to be broken during the transaction)], and correcting those violations so that the database is in a consistent state(i.e. no constraints) [("rows causing referential constraint violations, and new rows which are descendents of rows causing such violations are deleted")] teaches the aforementioned limitation.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Haderle's** would have allowed **Jacobs's** to provide a method for efficient processing of database constraints so that the order of transactions in terms of constraints are processed correctly, as noted by **Haderle** (Column 1, lines 44-51).

6. Claims 35, 40, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Jacobs et al.** (U.S. Patent 6,105,025) **Haderle et al.** (U.S. Patent 4,933,848) as applied to claims 31-33, 36-38, and 41-43 above, and further in view of **Jenkins** (U.S. Patent 5,899,993).

7. Regarding claims 35, 40, and 45, **Jacobs** and **Haderle** do not explicitly teach a constraint enforcer, method, and database system comprising:

A) wherein said constraints are selected from: primary keys, foreign keys, subset constraints, and exclude constraints.

Jenkins, however, teaches **"wherein said constraints are selected from: primary keys, foreign keys, subset constraints, and exclude constraints"** as "Not null, check, unique key, primary key, and foreign key constraints are defined in terms of queries on the database which are required to return an empty result. A constraint can be validated by running the constraint's defining query and confirming that the result is empty. The actual validation procedure may vary from implementation to implementation. For example, the validation procedure may be different for database systems that do not require processes to obtain a shared lock before reading data than

for database systems that require processes to obtain shared locks before reading data" (Column 7, lines 10-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Jenkins's** would have allowed **Jacobs's** and **Haderle's** to provide a method for allowing a variety of constraints to be enabled when database operations occur, as noted by **Jenkins** (Column 3, lines 30-34).

Response to Arguments

8. Applicant's arguments with respect to claims 31-33, 35-38, 40-43, and 45 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 5,706,494 issued to **Cochrane et al.** on 06 January 1998. The subject matter disclosed therein is pertinent to that of claims 31-33, 35-38, 40-43, and 45 (e.g., methods issue and enforce constraints on database systems).

U.S. Patent 6,453,314 issued to **Chan et al.** on 17 September 2002. The subject matter disclosed therein is pertinent to that of claims 31-33, 35-38, 40-43, and 45 (e.g., methods issue and enforce constraints on database systems).

U.S. Patent 5,408,657 issued to **Bigelow et al.** on 18 April 1995. The subject matter disclosed therein is pertinent to that of claims 31-33, 35-38, 40-43, and 45 (e.g., methods issue and enforce constraints on database systems).

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

March 5, 2008
/Mahesh H Dwivedi/
Examiner, Art Unit 2168
/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168